(19) JAPANESE PATENT OFFICE

PATENT ABSTRACTS OF JAPAN

(11) Publication number: **09074359 A**

(43) Date of publication of application: **18 . 03 . 97**

(51) Int. Cl **H03M 13/00**
G06F 11/10

(21) Application number: **07226414**

(22) Date of filing: **04 . 09 . 95**

(71) Applicant: **FUJITSU LTD**

(72) Inventor: **TAJIMA KAZUYUKI**
KAWAI MASAAKI
SHINOMIYA TOMOHIRO
ABIRU SETSUO
HIROTA MASAKI
MIYABE MASATAKE

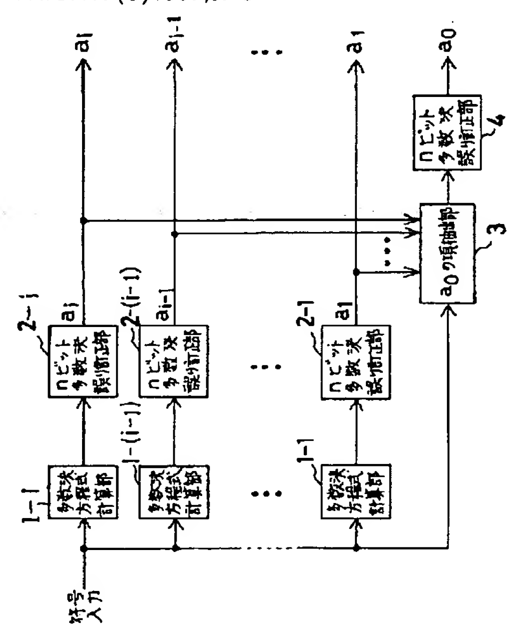
(54) **ERROR CORRECTION AND DECODING CIRCUIT**

(57) Abstract:

PROBLEM TO BE SOLVED: To enable a high-speed operation without using any ROM while using a simple logic circuit for decoding a Read-Muller code.

SOLUTION: Majority equation calculation parts from 1-1 to 1-i respectively composed of $2S^{-1}$ two-input adder circuits provided corresponding to respective terms of element information, a_0 to a_i [a_0 is least sightificant] are provided for parallelley inputting respective bit signals ($x_0 \ldots x_p$) of an input signal (x) so as to perform error correction and decoding by receiving the signal (x) for which the code length is 2S and the respective terms a_0 to a_i of element information are made into binary linear Read-Muller codes according to the expression $x=a_0v_0\ldots+a_1v_1+a_{i-1}v_{i-1}$ with v_0 to v_i as respective linear bases. For $2S^{-1}$ pieces of respective outputs from those parts, the error correction of n ($=2S^{-2}-1$) bits is performed at respective n-bit majority error correction parts from 2-1 to 2-i, codes a_1 to a_i are generated, the term of a_0 is extracted at a term extraction part 3 for a_0 by the input signal and the respective codes a_1 to a_i and next, error correction is performed by an n-bit majority error correction part 4.

COPYRIGHT: (C)1997,JPO



(19)日本国特許庁（ＪＰ）　(12) **公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－74359

(43)公開日　平成９年(1997)３月18日

(51)Int.Cl.[6]	識別記号	庁内整理番号	ＦＩ	技術表示箇所
Ｈ０３Ｍ　13/00			Ｈ０３Ｍ　13/00	
Ｇ０６Ｆ　11/10	３３０		Ｇ０６Ｆ　11/10	３３０Ｓ

審査請求　未請求　請求項の数５　ＯＬ　（全 23 頁）

(21)出願番号　特願平7－226414

(22)出願日　平成７年(1995)９月４日

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号

(72)発明者　田島　一幸
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　河合　正昭
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(74)代理人　弁理士　穂坂　和雄　（外２名）

最終頁に続く

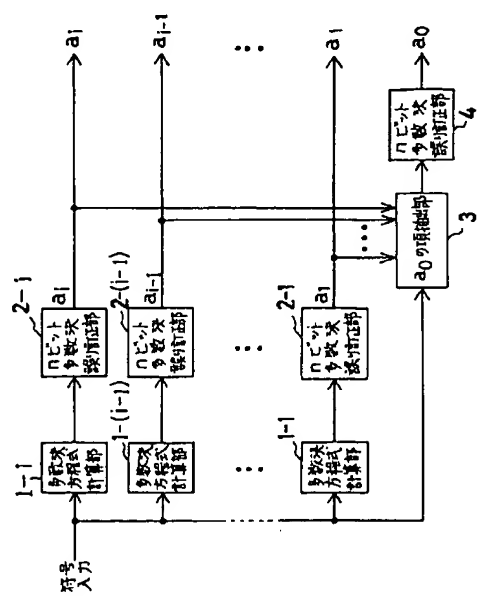
(54)【発明の名称】　誤り訂正復号回路

(57)【要約】

【課題】本発明は誤り訂正復号回路に関し，リード・マラー符号の復号を簡単な論理回路を用いてＲＯＭを使用せずに高速動作を可能にすることを目的とする。

【解決手段】符号長が 2^s で元の情報の各項 a_0 ～ a_i（a_0 が最下位）が v_0 ～ v_i を一次の各基底として式，$x = a_0 v_0 + a_1 v_1 \cdot\cdot + a_{i-1} v_{i-1}$ により２元一次のリード・マラー符号化された信号ｘを受け取って誤り訂正と復号を行うため，入力信号ｘの各ビット信号（$x_0 \cdots x_p$）が並列に入力され，a_1 ～ a_i の各項に対応して設けたそれぞれ２入力の 2^{s-1} 個の加算回路で構成された多数決方程式計算部を備える。そこからの 2^{s-1} 個の各出力は各ｎビット多数決誤り訂正部でｎ（$= 2^{s-2} - 1$）ビットの誤り訂正を行って符号 a_1 ～ a_i を発生し，a_0 の項抽出部で入力信号と各符号 a_1 ～ a_i により a_0 項を抽出し，次にｎビット多数決誤り訂正部で誤り訂正を行うよう構成する。

本発明の原理構成図



【特許請求の範囲】

【請求項1】　符号長が2^sで元の情報の各項a_0〜a_i（a_0が最下位）がv_0〜v_iを零次および一次の各基底として次の式

$$x=a_0 v_0+a_1 v_1+\cdots+a_{i-1} v_{i-1}+a_i v_i$$

により２元一次のリード・マラー符号化された信号xを受け取って誤り訂正と復号を行う誤り訂正復号回路において，入力信号xの各ビット信号（x_0，x_p：$p=2^s-1$）が並列に入力され，a_1〜a_iの各項に対応して設けられそれぞれ２入力の2^{s-1}個の加算回路で構成された多数決方程式計算部と，前記各多数決方程式計算部からの2^{s-1}個の出力が入力されてそれぞれnビット（$n=2^{s-2}-1$）の多数決誤り訂正出力を発生してそれぞれ符号a_1〜a_iの各出力を発生するnビット多数決誤り訂正部と，入力信号と前記各nビット多数決誤り訂正部の全ての出力とが入力されて符号a_0項を抽出するa_0の項抽出部と，その出力である2^s個の信号から誤り訂正したa_0を発生するnビット多数決誤り訂正部とで構成されることを特徴とする誤り訂正復号回路。

【請求項2】　符号長が2^sで元の情報の各項a_0〜a_i（a_0が最下位）の符号がv_0〜v_iを一次の各基底として次の式

$$x=a_0 v_0+(a_1+a_0)v_1+(a_2+a_0)v_2+\cdots(a_{i-1}+a_0)v_{i-1}+(a_i+a_0)v_i$$

により２元一次のリード・マラー符号化された信号xを受け取って誤り訂正と復号を行う誤り訂正復号回路において，請求項１に記載の各項a_1〜a_iに対応する前記多数決方程式計算部と，前記多数決方程式計算部に対応して設けられた各nビット多数決方程式計算部と，前記a_0の項抽出部とその出力が入力されて誤り訂正されたa_0を発生するnビット多数決誤り訂正部とを備え，前記a_1〜a_iに対応する各nビット多数決方程式計算部からの各出力と，前記a_0を発生するnビット多数決誤り訂正部の出力とが入力される各排他的論理和手段を設けたことを特徴とする誤り訂正復号回路。

【請求項３】　請求項１または２において，前記複数の多数決方程式計算部の出力を時分割で選択するセレクタと，前記セレクタの出力を受け取ってnビット多数決の誤り訂正を行い各項の出力を順番に発生する１個のnビット多数決誤り訂正部を設けたことを特徴とする誤り訂正復号回路。

【請求項４】　請求項１乃至３に記載の誤り訂正回路の入力側に，上記符号長が2^sのリード・マラー符号の上位ビットをラッチして前記誤り訂正回路へ出力する上位ラッチと，下位ビットをラッチする下位ラッチと，前記下位ラッチの出力と，符号長が2^{s-1}で１ビットを表し，各要素が互いに排他的な関係をもつ2^{s-1}多数決符号の情報“０”を表す符号とが入力される下位セレクタとを備え，符号長が2^sの２元一次のリード・マラー符号化された信号を復号する時，前記下位セレクタで下位ラッチの出力を選択して，前記上位ラインの出力と共に前記誤り訂正回路へ入力して復号と誤り訂正を行い，符号長が2^sの上位に前記2^{s-1}多数決符号が受信されると，前記下位セレクタを前記2^{s-1}多数決符号の情報“０”を選択するよう切り替えて，前記誤り訂正回路の中の上位ビットと下位ビットの排他的論理和を行う多数決方程式計算部の出力から前記2^{s-1}多数決符号の誤り訂正出力を得ることを特徴とする誤り訂正復号回路。

【請求項５】　請求項４において，前記上位ラッチの出力と符号長が2^{s-1}で１ビットを表し，各要素が互いに排他的な関係をもつ2^{s-1}多数決符号の情報“０”を表す符号とが入力されるセレクタと，下位ビットをラッチして誤り訂正回路へ出力する下位ラッチとを備え，符号長が2^sの下位に前記2^{s-1}多数決符号が受信されると，前記上位セレクタを前記2^{s-1}多数決符号の情報“０”を選択するよう切り替えることを特徴とする誤り訂正復号回路。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は多ビット誤りを訂正できるリード・マラー符号等の誤り訂正復号回路に関する。

【０００２】リード・マラー符号は，誤り検出，誤り訂正が可能な符号として符号化の技術において良く知られている。伝送装置のようにリード・マラー符号により信号を符号化してシリアル形式で伝送する場合，伝送速度が低いと受信側では復号回路を論理回路により構成するよりも回路規模を小さくすることができるＲＯＭを使用して誤り訂正を含む復号を行っていた。しかし，伝送速度が高速化すると，それに適応する高速ＲＯＭは高価であり，コスト・パフォーマンスの面で使用することができなかった。

【０００３】

【従来の技術】リード・マラー符号は，符号化の技術分野において一般に知られている。具体的な参考文献を挙げると，例えば，宮川洋，岩垂好裕，今井秀樹共著，コンピュータ基礎講座１８，「符号理論」昭晃堂（PP.168-176）がある。

【０００４】最初に，リード・マラー符号の符号化について説明する。リード・マラー符号は線型符号の一種で，一次の符号は陪直交符号であり，この発明に関係するのは二元一次の符号であるため，二元一次の符号について説明し，多元の場合は省略する。

【０００５】リード・マラー符号の定義の前提として，以下のベクトル積を定義する。

$a=(a_1, a_2, \cdots, a_n)$，$b=(b_1, b_2, \cdots, b_n)$

この二つのベクトルがあった時，ベクトル積を次のように定義する。

【０００６】

$c=ab=(a_1\ b_1,\ a_2\ b_2,\ \cdots\ a_n\ b_n)$

一次のリード・マラー符号の基底は以下のように定義される。但し，$n=2^s$とする。

【0007】

【数1】

$$v_0 = (v_0^0, v_0^1, \cdot\cdot\cdot\cdot, v_0^{n-1})$$

$$v_0^i = 1 \qquad i=0,1,2,\cdot\cdot\cdot,2^s-1$$

$$v_1 = (v_1^0, v_1^1, \cdot\cdot\cdot\cdot, v_1^{n-1})$$

$$v_1^i = 0 \qquad 0 \leqq i < 2^s-1$$

$$v_1^i = 1 \qquad \text{その他の } i$$

$$v_k \quad (v_k^0\ v_k^1, \cdot\cdot\cdot\cdot, v_k^{n-1})$$

$$v_k^i = 0 \qquad j\,(2^{s-k+1}) \leqq i < j\,(2^{s-k+1}) + 2^{s-k}$$

$$j=0,1,2,\cdot\cdot\cdot,2^s-1$$

$$v_k^i = 1 \qquad \text{その他の } i$$

【0008】高次の基底はこれら一次の基底のベクトル積で定義され，r次の基底は一次の基底r個のベクトル積で定義される。従って，符号長2^sのr次のリード・マラー符号はv_0，v_1，v_2，…v_sのr個までの積を基底とした$1+{}_sC_1+{}_sC_2+\cdots{}_sC_r$次元の線型ベクトル空間として定義される。例えば，S＝5の場合，符号長(n)は32となり，v_0及び一次の基底は以下のようになる。

【0009】

v_0 ＝(11111111111111111111111111111111)
v_1 ＝(00000000000000001111111111111111)
v_2 ＝(00000000111111110000000011111111)
v_3 ＝(00001111000011110000111100001111)
v_4 ＝(00110011001100110011001100110011)
v_5 ＝(01010101010101010101010101010101)

これを一次の符号として使用する場合は，情報を$(a_0, a_1, a_2, a_3, a_4, a_5)$とすると符号出力bは，

$$b=a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4 + a_5 v_5 \quad (1)$$

または，次のようになる。

$$b=a_0 v_0 + (a_1+a_0) v_1 + (a_2+a_0) v_2 + (a_3+a_0) v_3 + (a_4+a_0) v_4 + (a_5+a_0) v_5 \quad (2)$$

【0010】情報を$(a_0, a_1, a_2, a_3, a_4, a_5)$ ＝ (0, 0, 1, 0, 0, 1)とすると，$b=v_1+v_5$ ＝ (0101010101010101011010101010101010) となる。但し，二元なので，加算は排他的論理和になる。

【0011】次にリード・マラー符号の復号について説明する。リード・マラー符号の復号は，最高次の項から順に多数決判定法を用いて行う。上記の32ビットの例では式が長大となり説明を簡単にするため，8ビット（S＝3）を例として示し，説明の都合上最高次の項まで含む符号から復号する。

【0012】8ビットの場合の基底を全て示すと，次のv_0，v_1，v_2，v_3が一次，v_{12}，v_{13}，v_{23}が2次，v_{123}が3次（最高次）の式である。

v_0 ＝（11111111）
v_1 ＝（00001111）
v_2 ＝（00110011）
v_3 ＝（01010101）
$v_{12}=v_1 \cdot v_2$ ＝（00000011）
$v_{13}=v_1 \cdot v_3$ ＝（00000101）
$v_{23}=v_2 \cdot v_3$ ＝（00010001）
$v_{123}=v_1 \cdot v_2 \cdot v_3$ ＝（00000001）

符号出力bは，次のようになる。

【0013】$b=a_0 v_0 + a_1 v_1 + a_2 v_2 + a_3 v_3 + a_{12} v_{12} + a_{13} v_{13} + a_{23} v_{23} + a_{123} v_{123}$

ここで，v_0と他の基底との内積をとると，次のようになる。但し，1の加算により1の数が偶数の場合は0，奇数の場合は1になる。

【0014】$v_0 \cdot v_0=0$，$v_1 \cdot v_0=0$，$v_2 \cdot v_0=0$，$v_3 \cdot v_0=0$

$v_{12} \cdot v_0=0$，$v_{13} \cdot v_0=0$，$v_{23} \cdot v_0=0$

$v_{123} \cdot v_0=0+0+\cdot\cdot\cdot 0+1=1$

これは，符号長が変わっても同じ最高次の基底とv_0と

の内積のみ1でその他は0になることを表す。そこで，最高次の項の復号は，受信符号$x=(x_0, x_1, x_2, x_3, x_4, x_5, x_6, x_7)$としてxと$v_0$の内積をとればよい。

【0015】 $x\cdot v_0 = a_{123}\ v_{123}\cdot v_0 = a_{123} = x_0+x_1+x_2+x_3+x_4+x_5+x_6+x_7$

最高次の項は誤り検出も訂正もできないのでこの項の復号はこれで終わる。次に2次の項の復号を行うが，最高次の項の復号ができたので，xの項から最高次の項を取り除いた$x'=x-a_{123}\ v_{123}$をつくる。このx'とv_1の内積をとり，$v_i\cdot v_j=v_{ij}\cdot v_0$の関係を用いることにより，$a_{23}$が求められる。

【0016】 $x'\cdot v_1 = a_{23}v_{23}\cdot v_1 = a_{23}v_{123}\cdot v_0 = a_{23}=x'_4+x'_5+x'_6+x'_7$

ここで，$v_{23}(v_1+v_0)=v_{23}\cdot v_1+v_{23}\cdot v_0=v_{123}\cdot v_0$であるから，$x'(v_1+v_0)=a_{23}v_{123}\cdot v_0=a_{23}=x'_0+x'_1+x'_2+x'_3$となる。

【0017】このように，a_{23}が二つの式で求められ，この二つの式で多数決をとることで1ビットの誤りを検出できる。同様にa_{12}，a_{13}を求めることができる。二次の項の復号ができる。この後，一次の項の復号が，二次の復号と同様に行う。1次の復号では，xから最高次と二次の項を取り除いた次の式を作る。

【0018】 $x''=x-a_{123}\ v_{123}-(a_{12}v_{12}+a_{13}v_{13}+a_{23}v_{23})$

最初から二次以上の項が符号として使用されない場合は，x''を作る必要はなく，受信符号にそのまま処理を施すことができる。その方法は，二次の場合と同様で，以下の性質を利用して，次の4種類の式(1)～(4)で一次の項を求める。

【0019】

【数2】

$$v_1\cdot v_{23}=v_1\cdot v_{23}=v_{123}\cdot v_0 \quad (1)$$

$$v_1\cdot(v_2+v_0)v_3=v_1\cdot v_{23}+v_1\cdot v_3=v_{123}\cdot v_0 \quad (2)$$

$$v_1\cdot v_2(v_3+v_0)=v_1\cdot v_{23}+v_1\cdot v_2=v_{123}\cdot v_0 \quad (3)$$

$$v_1\cdot(v_2+v_0)(v_3+v_0)=v_1\cdot v_{23}+v_1\cdot v_2+v_1\cdot v_3+v_1\cdot v_0 =v_{123}\cdot v_0 \quad (4)$$

【0020】この性質から，a_1を求める場合，次の(5)～(8)に示す4つの式で多数決判定することにより，a_1について1ビットの誤り訂正復号ができる。

【0021】

【数3】

$$x''\cdot v_{23}=a_1v_{123}\cdot v_0=a_1=x''_3+x''_7 \quad (5)$$

$$x''\cdot(v_2+v_0)v_3=a_1v_{123}\cdot v_0=a_1=x''_1+x''_5 \quad (6)$$

$$x''\cdot v_2(v_3+v_0)=a_1v_{123}\cdot v_0=a_1=x''_2+x''_6 \quad (7)$$

$$x''(v_2+v_0)(v_3+v_0)=a_1v_{123}\cdot v_0=a_1=x''_0+x''_4 \quad (8)$$

【0022】同様にa_2，a_3についても求めることができる。なお，32ビットの場合には，16個ずつ5種類の多数決判定を行い，式が16個あるので7個までの誤りを訂正することができる。

【0023】最後にa_0が残るが，同様の方法で，x''から一次の項を取り除き，次の式を作る。

$x'''=x''-(a_1v_1+a_2v_2+a_3v_3)$

最後に残ったx'''はa_0v_0項だけなので，同様に書くとすれば，次の(9)～(16)で示す8個の式になる。

【0024】

【数4】

$$x'''\cdot v_{123}=a_0v_{123}\cdot v_0=a_0=x'''_7 \quad (9)$$

$$x'''\cdot(v_1+v_0)v_2v_3=a_0v_{123}\cdot v_0=a_0=x'''_3 \quad (10)$$

$$x'''\cdot v_1(v_2+v_0)v_3=a_0v_{123}\cdot v_0=a_0=x'''_5 \quad (11)$$

$$x'''\cdot v_1v_2(v_3+v_0)=a_0v_{123}\cdot v_0=a_0=x'''_6 \quad (12)$$

$$x'''\cdot(v_1+v_0)(v_2+v_0)v_3=a_0v_{123}\cdot v_0=a_0=x'''_1 \quad (13)$$

$$x'''\cdot(v_1+v_0)v_2(v_3+v_0)v_2=a_0v_{123}\cdot v_0=a_0=x'''_2 \quad (14)$$

$$x'''\cdot v_1(v_2+v_0)(v_3+v_0)=a_0v_{123}\cdot v_0=a_0=x'''_4 \quad (15)$$

$$x'''\cdot(v_1+v_0)(v_2+v_0)(v_3+v_0)=a_0v_{123}\cdot v_0=a_0=x'''_0 \quad (16)$$

【0025】この8個の式から多数決判定を行うことによりa₀が復号される。ここでは式が8個あるので多数

決判定すると３ビットまでの誤りを訂正できてしまうが，一次の符号も使用する場合，x'' を生成する段階で１ビットの誤りしか訂正できないので，それ以上の誤り訂正は不要である。この８個の式から分かるように，a_0 の復号は，x'' の各要素（x''_i，$i=0,1,\cdots,7$）でそのまま多数決をとっている。その他の符号長の場合も同様である。

【００２６】上記のようなリード・マラー符号を伝送装置で使用する場合，図１５に示す従来例の構成が採用されている。この従来例では，予め受信符号の 2^s 個のパターンについてソフトウェアシミュレーション等により復号を行い，受信符号と復号結果の対応関係を求め，その対応関係をＲＯＭにテーブルとして書き込んでおき，実際の復号を行う時，受信符号からテーブルを引いて復号結果を取り出す技術を用いている。そして，この図１５では，受信符号の上位側の符号を並列に展開してラッチ９０に，下位側の符号を並列に展開してラッチ９１にそれぞれラッチし，ＲＯＭ９２，９３はそれぞれ上位，下位の受信符号の各パターン（誤りを含む）をアドレスとして誤り訂正を含む復号結果が格納されている。セレクタ９４はＲＯＭ９２，９３の出力を切り換えて，順番に復号結果を取り出して情報出力を得る。

【００２７】

【発明が解決しようとする課題】従来は，リード・マラー符号の復号は規模が小さくなるＲＯＭを使用していたが，伝送速度が次第に高速化すると，それに合わせてＲＯＭも高速化する必要がある。しかし，高速のＲＯＭは高価であり，複数のＲＯＭを使用すると復号するための回路のコストが上るという問題があった。

【００２８】本発明はリード・マラー符号の復号を簡単な論理回路を用いてＲＯＭを使用せずに高速動作が可能な誤り訂正復号回路を提供することを目的とし，更にリード・マラー符号の半分の符号長で１ビットを表す多数符号を，リード・マラー符号の復号回路を使用して復号可能にすることを別の目的とする。

【００２９】

【課題を解決するための手段】図１は本発明の原理構成図である。図１の場合，原信号が a_0 ～ a_i である符号が二元で一次のリード・マラー符号化されて符号長が 2^s（$s=i$）である符号が使用され，その符号を復号するための原理構成を示す。また，符号化信号ｘは，上記式(1) の形式に対応する次の式により符号化されているものとする。

【００３０】$x=a_0 v_0 + a_1 v_1 + \cdots + a_{i-1} v_{i-1} + a_i v_i$

図１において，１－１～１－ｉは多数決方程式計算部，２－１～２－ｉはｎビット多数決誤り訂正部，３は a_0 の項抽出部，４は a_0 の項についてのｎビット多数決誤り訂正を行うｎビット多数決誤り訂正部である。

【００３１】リード・マラー符号化された符号入力は，並列に（または直列信号を並列信号に変換して）各多数決方程式計算部１へ入力される。各多数決方程式計算部１では，それぞれ入力する信号（ビット位置）が設定されており，それぞれ，2^{s-1} 個の排他的論理和を含む論理回路で構成される。各多数決方程式計算部１－１～１－ｉからの複数（2^{s-1} 個）の信号は，それぞれ対応するｎビット多数決誤り訂正部２－１～２－ｉへ入力する。ここで，各係数 a_1 ～ a_i について，$n=2^{s-2}-1$ 個の誤りを訂正する。これらの，一次の各係数 a_1 ～ a_i（情報ビット）はそれぞれ出力されると共に a_0 の項抽出部３へ入力される。ここで，$a_1 v_1 + a_2 v_2 + \cdots + a_{i-1} v_{i-1} + a_i v_i$ が作られ，これを受信した符号入力に加えることにより，a_0 の項を取り出す。但し，v_1 ～ v_i はリード・マラー符号の一次の基底である。この a_0 の項抽出部３からの 2^s 個の結果は，ｎビット多数決誤り訂正部４において $n=2^{s-2}-1$ 個の誤りを訂正して，a_0 の情報ビットが出力される。

【００３２】このような構成により，伝送速度が高速化しても，高価なＲＯＭを使用することなく，論理回路によりリード・マラー符号の誤り訂正回路を実現することができる。

【００３３】なお，リード・マラー符号の符号化が次の式により行われている場合，上記図１により復号した a_1 ～ a_i は（$a_1 + a_0$），（$a_2 + a_0$）・・・（$a_{i-1}+a_0$），（$a_i + a_0$）であるので，それぞれに a_0 を排他的論理和により加えて，a_1 ～ a_i を得る処理を行えばよい。

【００３４】$x=a_0 v_0 +(a_1 + a_0) v_1 +(a_2 + a_0) v_2 + \cdots (a_{i-1} + a_0) v_{i-1} +(a_i + a_0) v_i$

【００３５】

【発明の実施の形態】図２は具体的な構成例，図３は各多数決方程式計算部における計算の内容を示し，図４は各部の構成図，図５は a_0 の項抽出部の構成図である。

【００３６】図２において，１～４はそれぞれ上記図１の各符号に対応し，１－１～１－５は多数決方程式計算部，２－１～２－５は７ビット多数決誤り訂正部，３は a_0 の項抽出部，４は a_0 の項の７ビット多数決誤り訂正部である。

【００３７】この図２に示す構成は，リード・マラー符号の一次の符号で，Ｓ＝５の場合であり，符号長は 2^5 ＝３２（ビット）で x_{31} ～ x_0 で表し，情報ビット（復号結果）は６ビットで a_5 ～ a_0 で表す。

【００３８】各多数決方程式計算部１－１～１－５では，x_{31} ～ x_0 の符号入力に対し，a_1 ～ a_5 のそれぞれについて図３に示すような２つのビット入力に対し１６個（$=2^{5-1}$）の加算（二元であるため排他的論理和）を行う。図４のＡ．に情報ビット a_5 に対応する多数決方程式計算部（図２の１－５）の構成を示す。このように１６個の排他的論理和回路において，それぞれ決

められた２つの符号ビットについて排他的論理和を行い，１６個の出力は，それぞれ７ビット多数決誤り訂正部２－１～２－５へ入力される。７ビット多数決誤り訂正部の構成は，図４のＢ．に示され，１６個の信号は，８個設けられた２入力の１ビット加算器２０に順番に入力される。各１ビット加算器２０で“１”を加算し，各１ビット加算器２０の加算結果（２ビットの出力となる）として８個の出力が発生する。これらは，次に４個設けられた２入力の２ビット加算器２１に順番に入力され，それぞれ加算が行われる。この加算結果（３ビットになる）は次に２個設けられた３ビット加算器２２へ順次入力される。この加算結果（４ビットになる）は，次の２入力の４ビット加算器２３へ入力されて，加算結果（５ビットになる）はコンパレータ２４へ入力される。

【００３９】この場合，符号長が2^5であるから，$n=2^{5-2}-1=7$個の誤り訂正をするもので，１６個の入力の中で“１”が７個以下であれば“０”，“０”が７個以下であれば“１”に復号される。すなわち，図４のＢ．のコンパレータ２４は，４ビット加算器２３の加算結果（“１”の個数を表す）を数値“８”と比較し，一致する場合は，復号失敗の信号を出力し，８より多い場合は$a_i=1$（ｉは1～5の一つ）を表す信号，８より小さい場合は$a_i=0$を表す信号を出力する。論理回路２５は，コンパレータ２４の出力を論理信号に変換する回路であり，$a_i=1$の場合は論理“１”，$a_i=0$の場合は論理“０”を発生する。

【００４０】次に図５に示すa_0の項抽出部の構成により復号された一次の係数a_1～a_5を用いてa_0の項が抽出される。この場合，次の式の値を求め，受信符号（符号入力）を加える（減算と同じ）ことにより得る。

【００４１】

$$a_1 v_1 + a_2 v_2 + a_3 v_3 + a_4 v_4 + a_5 v_5$$

ここで，v_1～v_5はリード・マラー符号の一次の基底であり，３２ビットの場合上記に記載したような値である。v_1の例により説明すると，

v_1 ＝(00000000000000001111111111111111)

であり，上記の$a_1 v_1$の項により，係数a_1は，入力符号のx_{15}～x_{00}の各ビット（v_1が“１”であるビット位置）が加算される。同様に係数a_2は，v_2の“１”になっている位置の入力符号が加算される。係数a_3～a_5についても，v_3～v_5の値に対応して図５のように入力される。

【００４２】このように，a_0を表す３２個の排他的論理和の出力が発生すると，次に７ビット多数決誤り訂正部４において，多数決誤り訂正を行う。この７ビット多数決誤り訂正部４も，３２ビット符号の場合７ビットの誤りまでしか訂正できないので，a_0の多数決復号においても７ビット誤り訂正復号を行う。ここで行う７ビット多数決誤り訂正は，上記の７ビット多数決誤り訂正部２－１～２－５と機能は同一であるが，入力ビットの数は２倍ある。

【００４３】上記の復号は，符号化時に，上記の式(1)に基づいて行った場合であるが，上記の式(2)により符号化した場合（各基底にa_0が含まれる場合）の構成を図６に示す。すなわち，図６は各基底にa_0を含む符号を復号する場合の構成であり，この場合，符号ｂは次のように符号化され各基底v_0～v_5の係数にa_0を含んでいる。

【００４４】$b=a_0 v_0 + (a_1 + a_0) v_1 + (a_2 + a_0) v_2 + (a_3 + a_0) v_3 + (a_4 + a_0) v_4 + (a_5 + a_0) v_5$

図６において，１０－１～１０－ｉは多数決方程式計算部，１１－１～１１－ｉはｎビット多数決誤り訂正部，１２はa_0の項抽出部，１３はa_0の項のｎビット多数決誤り訂正部，１４－１～１４－ｉは排他的論理和回路である。

【００４５】この構成では，各多数決方程式計算部１０－１～１０－ｉは，それぞれ符号入力について，上記図１と同様に2^{5-1}個の入力について多数決方程式計算を行い，それぞれの出力がｎビット多数決誤り訂正部１１－１～１１－ｉで$2^{5-2}-1$個の多数決誤り訂正を行い，出力としてa_1+a_0，a_2+a_0，…，$a_{i-1}+a_0$，a_i+a_0を出力する。これらの出力は，a_0の項抽出部１２へ入力され，ここでa_0を表す2^5個の信号が発生し，ｎビット多数決誤り訂正部１３で誤り訂正が行われて，a_0が出力される。このa_0を各排他的論理和回路１４－１～１４－ｉへ入力する。この回路は上記図４のＣ．に示す後処理回路を構成し，a_i+a_0の入力とa_0の入力を排他的論理和回路へ入力すると，実質的に減算が行われてa_iが出力される。

【００４６】こうして，各ｎビット多数決誤り訂正部１１－１～１１－ｉの出力からa_0を除いた各情報a_1～a_iが発生する。図７は図１の構成においてｎビット多数決誤り訂正部を共通化した構成である。図７において，２は共通のｎビット多数決誤り訂正部，５はセレクタであり，他の１－１～１－ｉ，３，４の各符号は上記図１の同一符号と同様である。

【００４７】図７の場合，各多数決方程式計算部１－１～１－ｉの出力は，セレクタ５により時分割式に順番にｎビット多数決誤り訂正部２へ供給され，各出力はa_0の項抽出部３へ順番に入力する。a_0の項抽出部３はそれらを保持（ラッチ）して，a_0の項を抽出し，ｎビット多数決誤り訂正部４で誤り訂正を行う。

【００４８】図８は図６の構成においてｎビット多数決誤り訂正部を共通化した構成である。図８において，１１は共通のｎビット多数決誤り訂正部，１４はセレクタであり，他の１０－１～１０－ｉ，１２，１３，１４－１～１４－ｉの各符号は上記図６の同一符号と同様である。

【００４９】この動作も，上記図７と同様にセレクタ１

４により各多数決方程式計算部１０－１～１０－ｉの出力は，時分割式に順番にｎビット多数決誤り訂正部１１へ供給され，各出力はa_0の項抽出部１２へ供給されてa_0の項が抽出されると共に，排他的論理和回路１４－１～１４－ｉへ供給されて後処理（a_0を除く）が行われる。

【００５０】次に本発明によるリード・マラー符号の復号回路を，情報１ビットで符号長がリード・マラー符号の半分の長さを持つ多数決符号の復号に適用することができる。その原理と構成を以下に説明する。

【００５１】リード・マラー符号は符号長を2^sとすると，2^{s-2}－１個の誤りを訂正できる。つまり符号間のハミング距離は2^{s-1}個である。この時符号長を2^{s-1}で2^{s-2}－１個の誤りを訂正できるのは，１ビットの情報を符号長2^{s-1}の２個の符号に符号化し，一方の符号が他方の符号の各要素を反転した関係になっている場合ということになる。この符号を，以下，情報１ビットで符号長が2^{s-1}の多数決符号または2^{s-1}多数決符号という。

【００５２】例えば，Ｓ＝５の場合（2^{s-1}＝１６），１ビットが
(0000111100001111）と（1111000011110000）や(0101101001011010）と（1010010110100101）等のような符号になる。なお,これに対応するリード・マラー符号の符号長は2^s（＝３２）としているので，符号長は2^{s-1}になる。

【００５３】具体的に，Ｓ＝５の場合に，上記の前者の符号使用して以下のように符号化することができる。
情報＝０の時（0000111100001111）に符号化し，
情報＝１の時（1111000011110000）に符号化する。

【００５４】これを復号する場合，受信符号と情報＝０を符号化した符号（ここでは，0000111100001111）の各要素同士の排他的論理和をとって，その結果の各要素で多数決をとればよい。この場合，１が2^{s-2}＋１個以上（この例では９個以上）の場合は１に復号され，１が2^{s-2}－１個以下（この例では７個以下）であれば0に復号される。1が2^{s-2}個（この例では８個）のときは訂正不可となる。

【００５５】情報０の（0000111100001111）と情報０の（0000111100001111）の排他的論理和は，(0000000000000000)で，結果の中に1が０個あるので０に復号される。もし（0000111100001111）に１～７ビットの誤りが入ると，排他的論理和の中に１が１～７個見つかるが，７個以下なので０に復号される。逆に情報１の（1111000011110000）と情報０の(0000111100001111)の排他的論理和は，(1111111111111111)で，１が１６個あり，１に復号される。同様に（1111000011110000）に１～７ビットの誤りが入ると，排他的論理和の中に１が１５～９個みつかるが，９個以上なので１に復号される。

【００５６】これは情報ｉを受信符号を（x_{15} x_{14} x_{13} x_{12}…x_2　x_1　x_0）として，以下の多数決方程式を計算して７ビット多数決誤り訂正を行う処理に等しい。

$i=0+x_{15}$，$i=0+x_{14}$，$i=0+x_{13}$，$i=0+x_{12}$，$i=1+x_{11}$
$i=1+x_{10}$，$i=1+x_9$，$i=1+x_8$，$i=0+x_7$，$i=0+x_6$
$i=0+x_5$，$i=0+{}_{4\ 9}$，$i=1+x_3$，$i=1+x_2$，$i=1+x_1$
$i=1+x_0$

一方，リード・マラー符号の多数決方程式計算部の計算式が図５に示されている。この中の符号a_1の計算式を参照すると，受信符号の上位１６ビットと下位１６ビットの排他的論理和になっている。この符号長が2^sの場合でも，同様に受信符号の上位2^{s-1}ビットと下位2^{s-1}ビットの排他的論理和になる。

【００５７】そこで，受信符号の上位１６ビット入力として（x_{31} x_{30}…x_{18} x_{17} x_{16}）＝（0000111100001111 0000）を，下位１６ビットとして受信符号（x_{15} x_{14} x_{13} x_{12}…x_2　x_1　x_0）を入力すれば，a_1の出力として多数決復号された結果を出力することができる。この場合，上位と下位は逆でも同様である。また，これを一般化すると，受信符号入力の上位2^{s-1}ビットか，下位2^{s-1}ビットに情報＝０を符号化した符号を入力し，残りの下位か上位の2^{s-1}ビットに受信符号を入力すると，a_1の出力に多数決復号された結果が出力される。

【００５８】また，多数決方程式計算部の計算式は受信符号の二つのビットを取り出して排他的論理和をとっているので，ビットの入れ替えを行えば，a_1以外のa_2やa_3…の復号部でも復号は可能である。

【００５９】上記の原理を実現する2^{s-1}多数決符号とリード・マラー符号を復号する基本構成を図９に示し，図９のタイミングチャートを図１０に示す。図９において，３０は上位ビットを格納する上位ラッチ，３１は下位ビットを格納する下位ラッチ，３２は下位セレクタ，３３はリード・マラー符号復号回路である。リード・マラー符号回路３３は，上記図１，図２，図６～図８の何れかの回路で構成することができる。また，下位セレクタ３２はモード選択信号により切替えられ，符号長2^sのリード・マラー符号を復号する時は下位ラッチ３１を選択し，2^{s-1}多数決符号を復号する時は情報＝０の符号化した符号の方を選択する。2^{s-1}多数決符号を復号する場合は，リード・マラー符号復号回路３３のa_1の出力に多数決復号された結果が出力される。

【００６０】図１０により図９の動作タイミングを説明すると，受信符号はａ．に示すようにリード・マラー（ＲＭ符号と略して表示）(1) の上位，下位，2^{s-1}多数決符号，ＲＭ符号(2) の上位，下位，…の順に入力される。まず，ｂ．に示すように上位ラッチ３０でＲＭ符号(1) の上位がラッチされる。次にｃ．に示すように下位ラッチ３１にＲＭ符号(1) の下位がラッチされる。下

位セレクタ３２はｅ．に示す区間１ではｄ．に示すように下位ラッチ３１の方に開いており，区間１でＲＭ符号化(1) の上位と下位が揃うので，ＲＭ符号(1) が復号される。

【００６１】次に2^{s-1} 多数決符号が上位ラッチ３０にラッチされる。この時，下位セレクタ３２を2^{s-1} 多数決符号の情報＝０の符号の方に開いておくと，区間２では2^{s-1} 多数決符号が復号される。その後，上位ラッチ３０にＲＭ符号(2) の上位がラッチされる。次に下位ラッチ３１にＲＭ符号(2) の下位がラッチされると，下位セレクタ３２が下位ラッチ３１の方に開き，区間３ではＲＭ符号(2) が復号される。なお，2^{s-1} 多数決符号の復号出力は，リード・マラー符号復号回路３３の一つの符号出力（例えばa_1 ）として得られる。

【００６２】図１１は2^{s-1} 多数決符号の復号を含む復号回路の実施例１の構成図である。図１１において，多数決方程式計算部１－１～１－ｉ，ｎビット多数決誤り訂正部２－１～２－ｉ，a_0の項抽出部３及びｎビット多数決誤り訂正部４の各部は，上記図１と同様であり，入力側の構成は上記図９と同様に，受信符号上位ビットの入力と，受信符号下位ビットまたは2^{s-1} 多数決符号の情報＝０の符号が入力されるセレクタ６（図９の下位セレクタ３２に対応）を備えている。なお，この構成では，受信符号上位ビット及び受信符号下位ビットは図９のようなラッチから出力されてもよい。

【００６３】この図１１の動作は，上記図９，図１０に説明したのと同様にリード・マラー符号の復号（上記図６と同じ）と，2^{s-1} 多数決符号の復号を行う。なお，2^{s-1} 多数決符号の復号出力は，ｎビット多数決誤り訂正部２－１から出力（a_1 の出力）して発生され，図には「他の符号」と表示されている。

【００６４】図１２は2^{s-1} 多数決符号の復号を含む復号回路の実施例２の構成図である。図１２の場合，2^{s-1} 多数決符号の復号を上記図６の構成に適用したものであり，図１２の多数決方程式計算部１０－１～１０－ｉ，ｎビット多数決誤り訂正部１１－１～１１－ｉ，a_0 の項抽出部１２及びｎビット多数決誤り訂正部１３，排他的論理和１４－１～１４－ｉの各部は，上記図６と同様であり，入力側に上記図１１と同様の入力が供給されるセレクタ１５が設けられている。

【００６５】この図１２の構成でも，上記図６と同様のリード・マラー符号の復号と，2^{s-1} 多数決符号の復号を，上記図９，図１０について説明した方法により行い，2^{s-1} 多数決符号の復号出力は，上記図１１と同様にｎビット多数決誤り訂正部２－１からの出力（a_1 の出力）として発生する。

【００６６】図１３は2^{s-1} 多数決符号の復号を含む復号回路の実施例３の構成図である。この構成は，上記図７に示す構成により2^{s-1} 多数決符号の復号を可能とするものであり，図７の入力側に上記図１１，図１２と同様に，受信符号上位ビットの入力と，受信符号下位ビットまたは2^{s-1} 多数決符号の情報＝０の符号が入力されるセレクタ６を備えている。この図１３の構成の動作は，上記図１１と同様でありリード・マラー符号の復号と，2^{s-1} 多数決符号の復号を行い，2^{s-1} 多数決符号の復号出力は，共通回路であるｎビット多数決誤り訂正部２からのa_1の出力として発生する。

【００６７】図１４は2^{s-1} 多数決符号の復号を含む復号回路の実施例４の構成図である。この構成は，上記図８に示す構成により2^{s-1} 多数決符号の復号を可能とするものであり，図８の入力側に上記図１２と同様に，受信符号上位ビットの入力と，受信符号下位ビットまたは2^{s-1} 多数決符号の情報＝０の符号が入力されるセレクタ１５を備えている。この図１４の構成の動作は，上記図１１～図１３と同様でありリード・マラー符号の復号と，2^{s-1} 多数決符号の復号を行い，2^{s-1}多数決符号の復号出力は，共通回路であるｎビット多数決誤り訂正部１１からのa_1 の出力として発生する。

【００６８】上記の説明では入力信号がシリアルの信号を並列信号に変換した後復号回路へ入力するものとして説明したが，最初から並列な信号として入力しても動作することができる。

【００６９】

【発明の効果】本発明によればリード・マラー符号の復号を簡単な論理回路を組み合わせることにより構成することができ，符号信号がが高速化しても対応することができる。

【００７０】また，ｎビット多数決誤り訂正部を時分割で動作させる構成により回路規模を小さくし，回路のコストを低下させることができる。次に１ビットを2^{s-1}の符号長で表す多数決符号を，符号長2^s のリード・マラー符号の復号回路を用いて復号することを可能とし，重要なビット情報が確実に受け取ることが可能となる。

【図面の簡単な説明】

【図１】本発明の原理構成図である。

【図２】具体的な構成例を示す図である。

【図３】各多数決方程式計算部における計算の内容を示す図である。

【図４】各部の構成図である。

【図５】a_0の項抽出部の構成図である。

【図６】各基底にa_0を含む符号を復号する場合の構成を示す図である。

【図７】図１の構成においてｎビット多数決誤り訂正部を共通化した構成を示す図である。

【図８】図６の構成においてｎビット多数決誤り訂正部を共通化した構成を示す図である。

【図９】2^{s-1} 多数決符号とリード・マラー符号を復号する基本構成を示す図である。

【図１０】図９のタイミングチャートを示す図である。

【図１１】2^{s-1} 多数決符号の復号を含む復号回路の実

施例１の構成図である。

【図１２】 2^{s-1} 多数決符号の復号を含む復号回路の実施例２の構成図である。

【図１３】 2^{s-1} 多数決符号の復号を含む復号回路の実施例３を構成図である。

【図１４】 2^{s-1} 多数決符号の復号を含む復号回路の実施例４を構成図である。

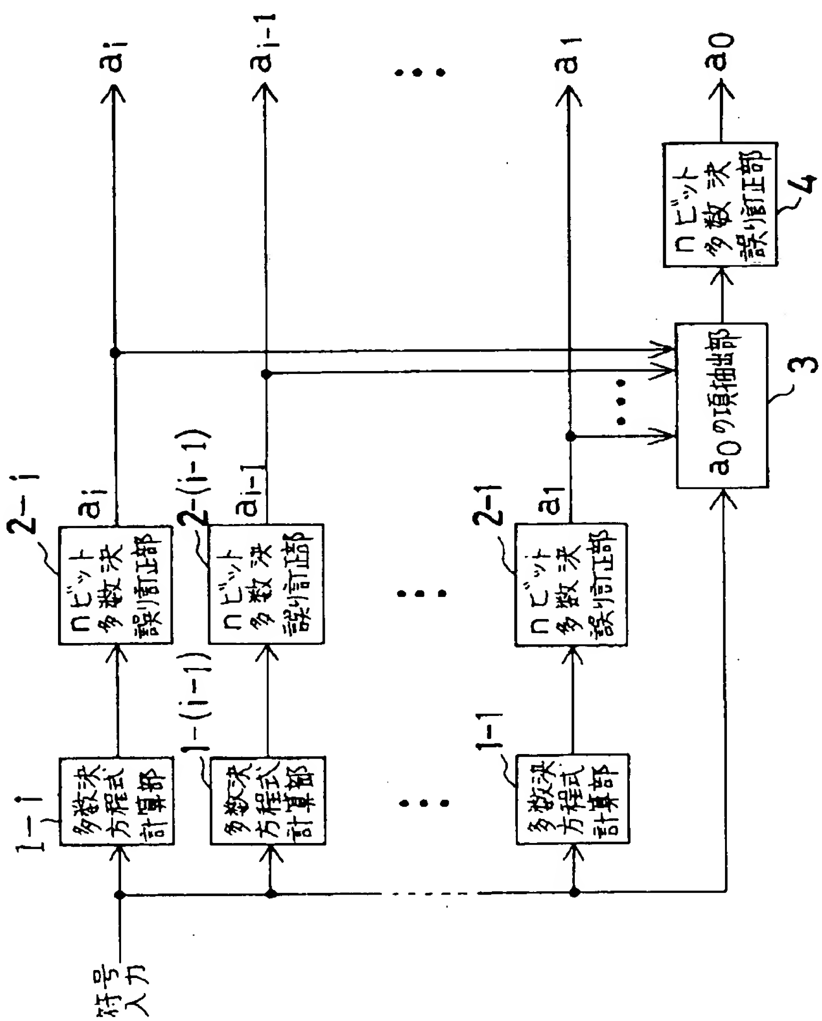
【図１５】従来例の構成図である。

【符号の説明】

１－１～１－ｉ	多数決方程式計算部
２－１～２－ｉ	ｎビット多数決誤り訂正部
３	a_0 の項抽出部
４	a_0 のｎビット多数決誤り訂正部

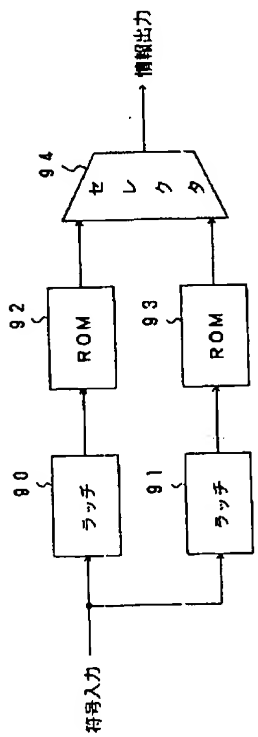
【図１】

本発明の原理構成図



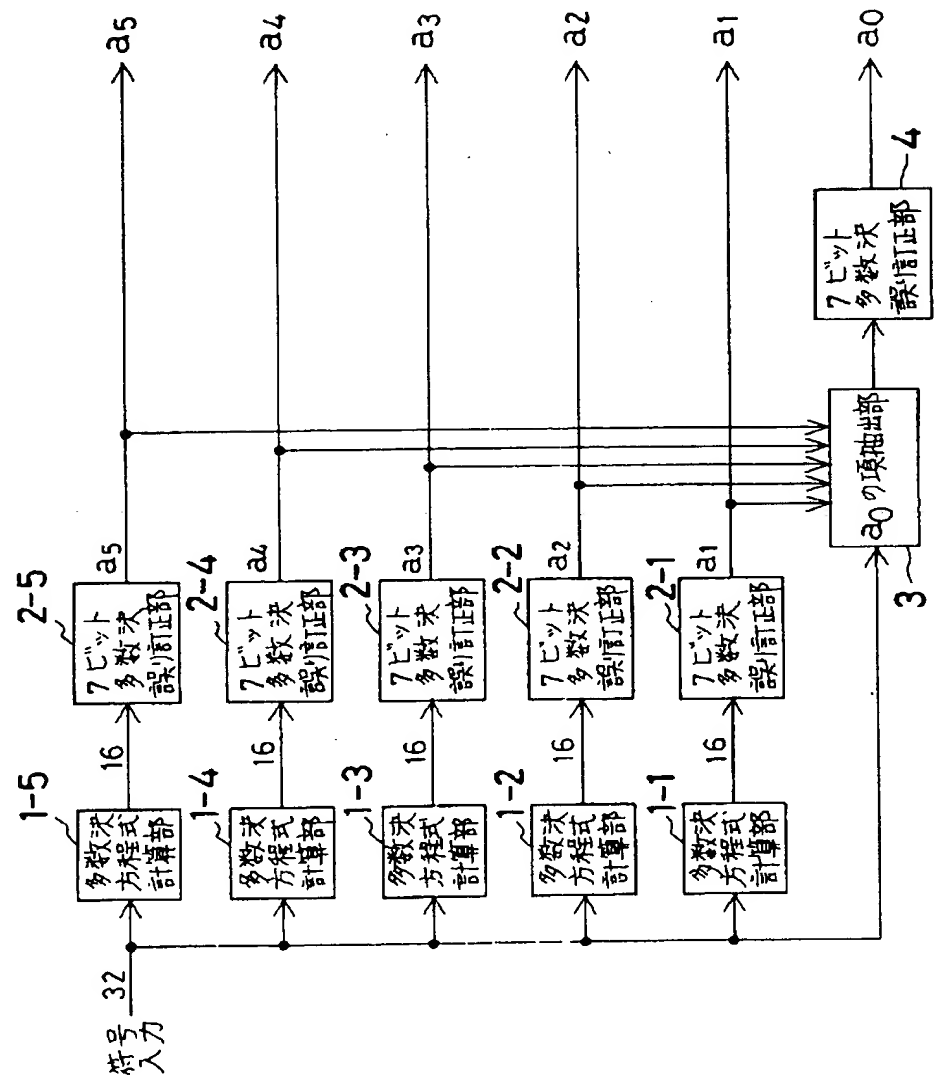
【図１５】

従来例の構成図



【図２】

具体的な構成例


符号入力
32
多数決方程式計算部
1-1
1-2
1-3
1-4
1-5
16
7ビット多数決誤り訂正部
2-1
2-2
2-3
2-4
2-5
a_1
a_2
a_3
a_4
a_5
a_0の項抽出部
3
4
a_0

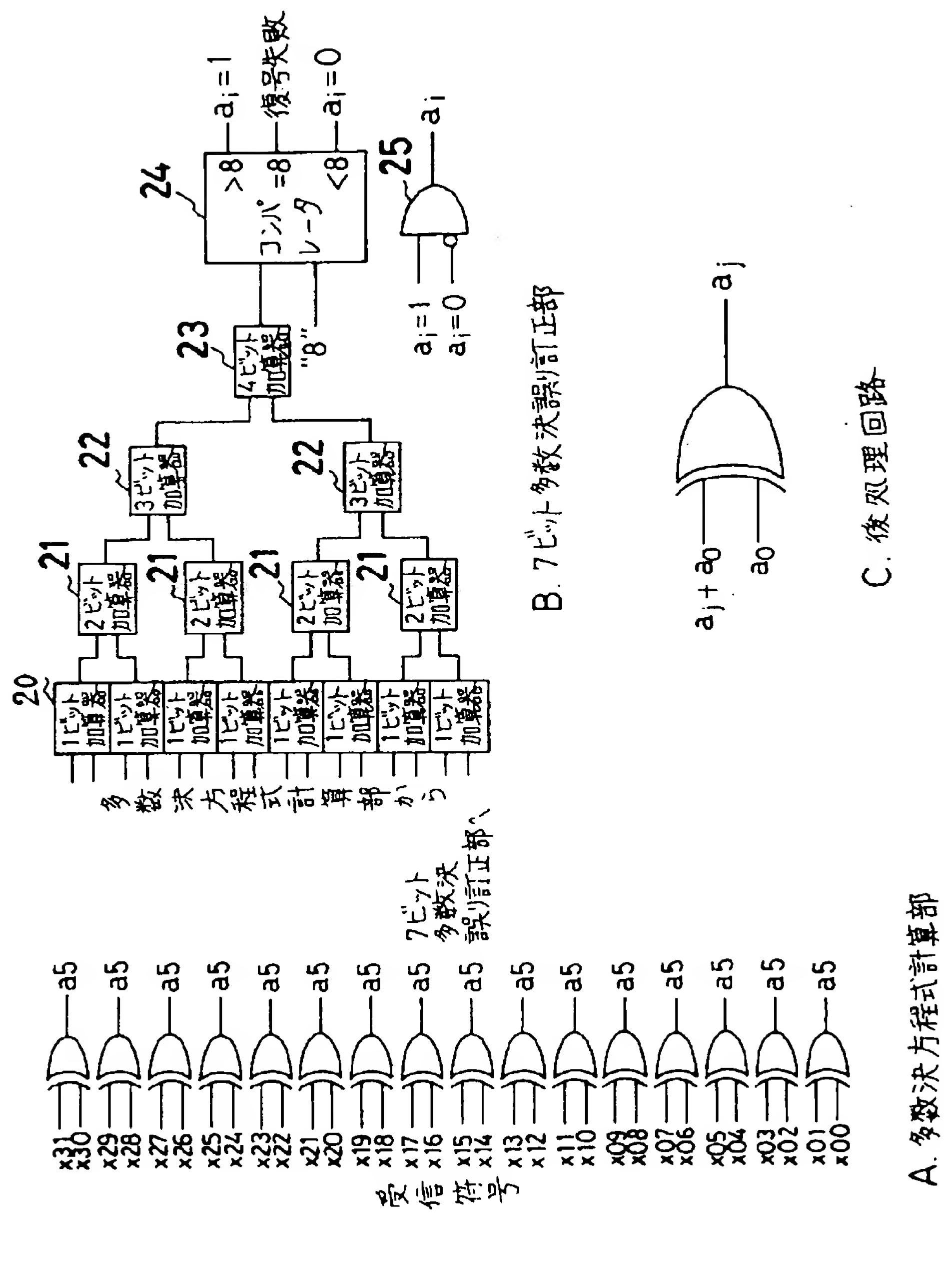
【図3】

各多数決方程式計算部における計算の内容

$a_5 = x_{31} + x_{30}$	$a_4 = x_{31} + x_{29}$	$a_3 = x_{31} + x_{27}$	$a_2 = x_{31} + x_{23}$	$a_1 = x_{31} + x_{15}$
$a_5 = x_{29} + x_{28}$	$a_4 = x_{30} + x_{28}$	$a_3 = x_{30} + x_{26}$	$a_2 = x_{30} + x_{22}$	$a_1 = x_{30} + x_{14}$
$a_5 = x_{27} + x_{26}$	$a_4 = x_{27} + x_{25}$	$a_3 = x_{29} + x_{25}$	$a_2 = x_{29} + x_{21}$	$a_1 = x_{29} + x_{13}$
$a_5 = x_{25} + x_{24}$	$a_4 = x_{26} + x_{24}$	$a_3 = x_{28} + x_{24}$	$a_2 = x_{28} + x_{20}$	$a_1 = x_{28} + x_{12}$
$a_5 = x_{23} + x_{22}$	$a_4 = x_{23} + x_{21}$	$a_3 = x_{23} + x_{19}$	$a_2 = x_{27} + x_{19}$	$a_1 = x_{27} + x_{11}$
$a_5 = x_{21} + x_{20}$	$a_4 = x_{22} + x_{20}$	$a_3 = x_{22} + x_{18}$	$a_2 = x_{26} + x_{18}$	$a_1 = x_{26} + x_{10}$
$a_5 = x_{19} + x_{18}$	$a_4 = x_{19} + x_{17}$	$a_3 = x_{21} + x_{17}$	$a_2 = x_{25} + x_{17}$	$a_1 = x_{25} + x_{9}$
$a_5 = x_{17} + x_{16}$	$a_4 = x_{18} + x_{16}$	$a_3 = x_{20} + x_{16}$	$a_2 = x_{24} + x_{16}$	$a_1 = x_{24} + x_{8}$
$a_5 = x_{15} + x_{14}$	$a_4 = x_{15} + x_{13}$	$a_3 = x_{15} + x_{11}$	$a_2 = x_{15} + x_{7}$	$a_1 = x_{23} + x_{7}$
$a_5 = x_{13} + x_{12}$	$a_4 = x_{14} + x_{12}$	$a_3 = x_{14} + x_{10}$	$a_2 = x_{14} + x_{6}$	$a_1 = x_{22} + x_{6}$
$a_5 = x_{11} + x_{10}$	$a_4 = x_{11} + x_{9}$	$a_3 = x_{13} + x_{9}$	$a_2 = x_{13} + x_{5}$	$a_1 = x_{21} + x_{5}$
$a_5 = x_{9} + x_{8}$	$a_4 = x_{10} + x_{8}$	$a_3 = x_{12} + x_{8}$	$a_2 = x_{12} + x_{4}$	$a_1 = x_{20} + x_{4}$
$a_5 = x_{7} + x_{6}$	$a_4 = x_{7} + x_{5}$	$a_3 = x_{7} + x_{3}$	$a_2 = x_{11} + x_{3}$	$a_1 = x_{19} + x_{3}$
$a_5 = x_{5} + x_{4}$	$a_4 = x_{6} + x_{4}$	$a_3 = x_{6} + x_{2}$	$a_2 = x_{10} + x_{2}$	$a_1 = x_{18} + x_{2}$
$a_5 = x_{3} + x_{2}$	$a_4 = x_{3} + x_{1}$	$a_3 = x_{5} + x_{1}$	$a_2 = x_{9} + x_{1}$	$a_1 = x_{17} + x_{1}$
$a_5 = x_{1} + x_{0}$	$a_4 = x_{2} + x_{0}$	$a_3 = x_{4} + x_{0}$	$a_2 = x_{8} + x_{0}$	$a_1 = x_{16} + x_{0}$

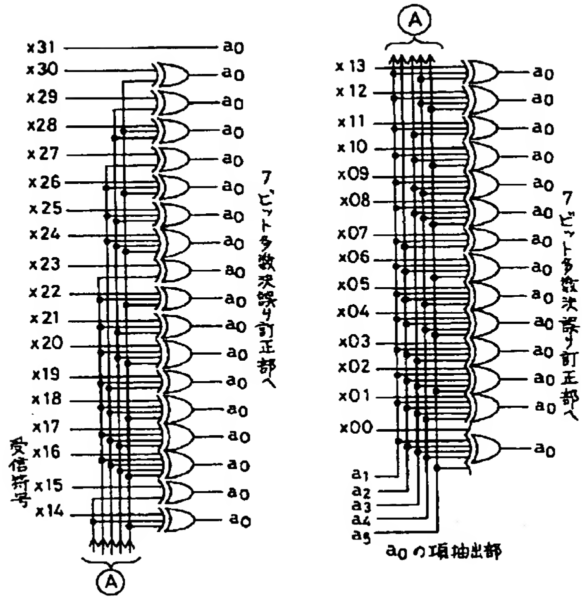
【図4】

各部の構成図


受信符号
多数決方程式計算部から
7ビット多数決誤り訂正部へ
1ビット加算器
2ビット加算器
3ビット加算器
4ビット加算器
コンパレータ
"8"
>8
=8
<8
$a_i=1$
復号失敗
$a_i=0$
a_i
a_j+a_0
a_0
a_j
A. 多数決方程式計算部
B. 7ビット多数決誤り訂正部
C. 復号処理回路

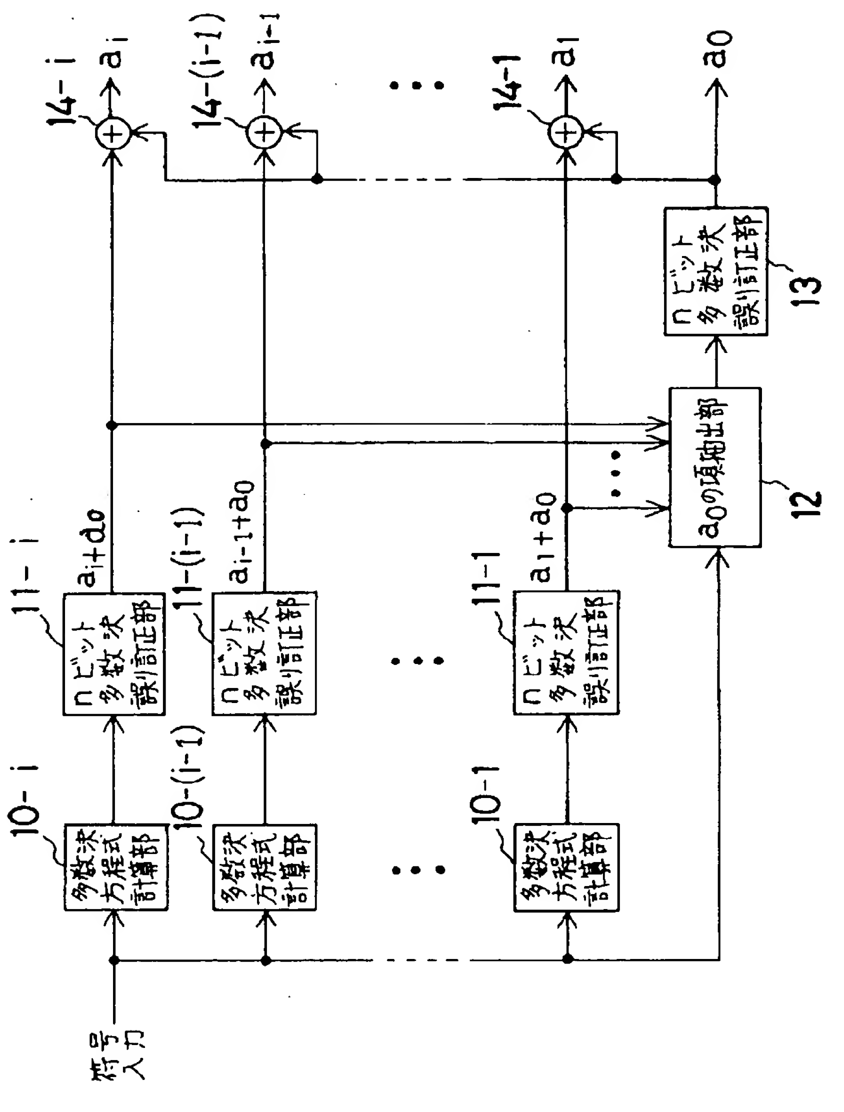
【図５】

a_0の項抽出部の構成図



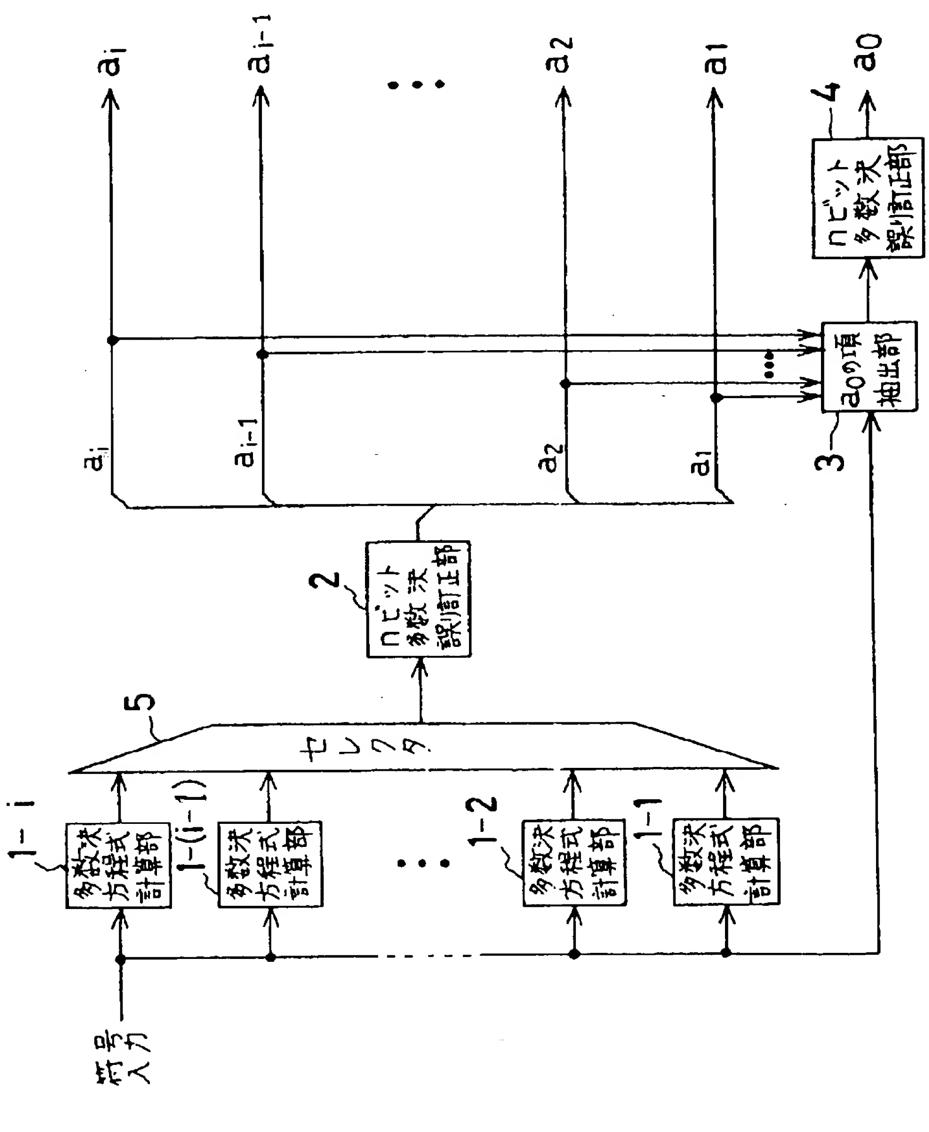
【図6】

各基底に a_0 を含む符号を復号する場合の構成



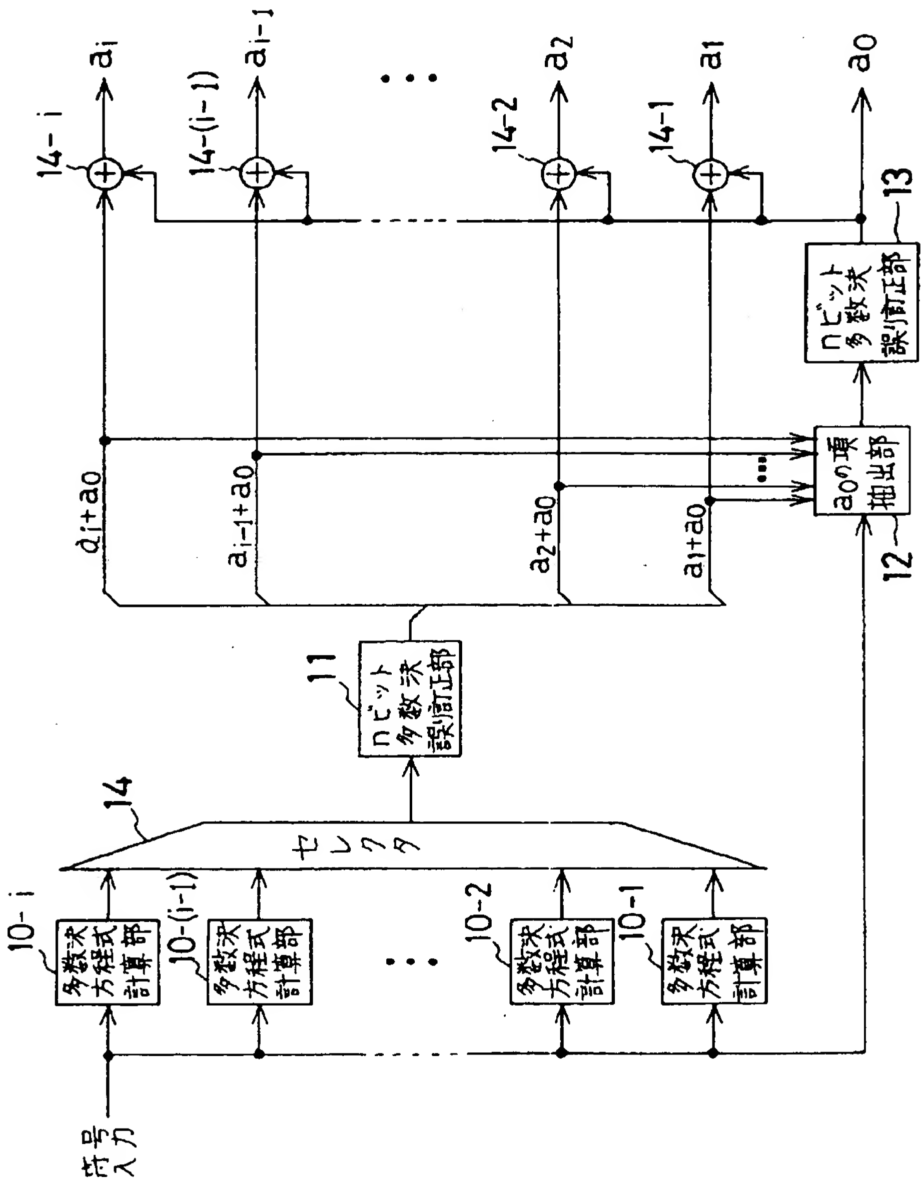
【図7】

図1の構成においてnビット多数決誤り訂正部を共通化した構成図



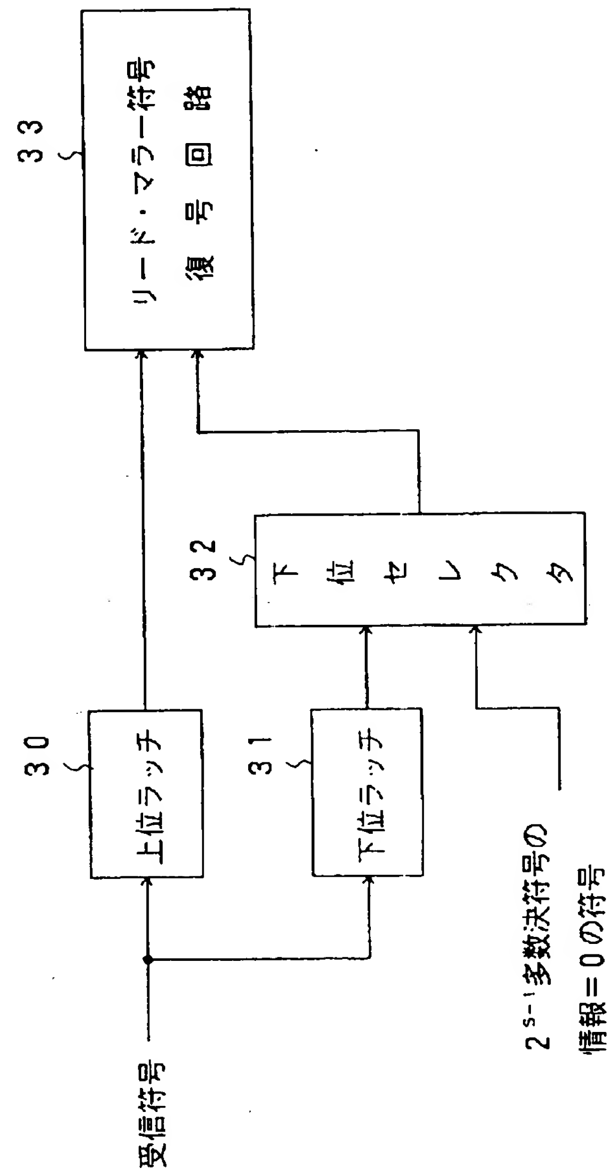
【図８】

図６の構成においてｎビット多数決誤り訂正部を
共通化した構成図



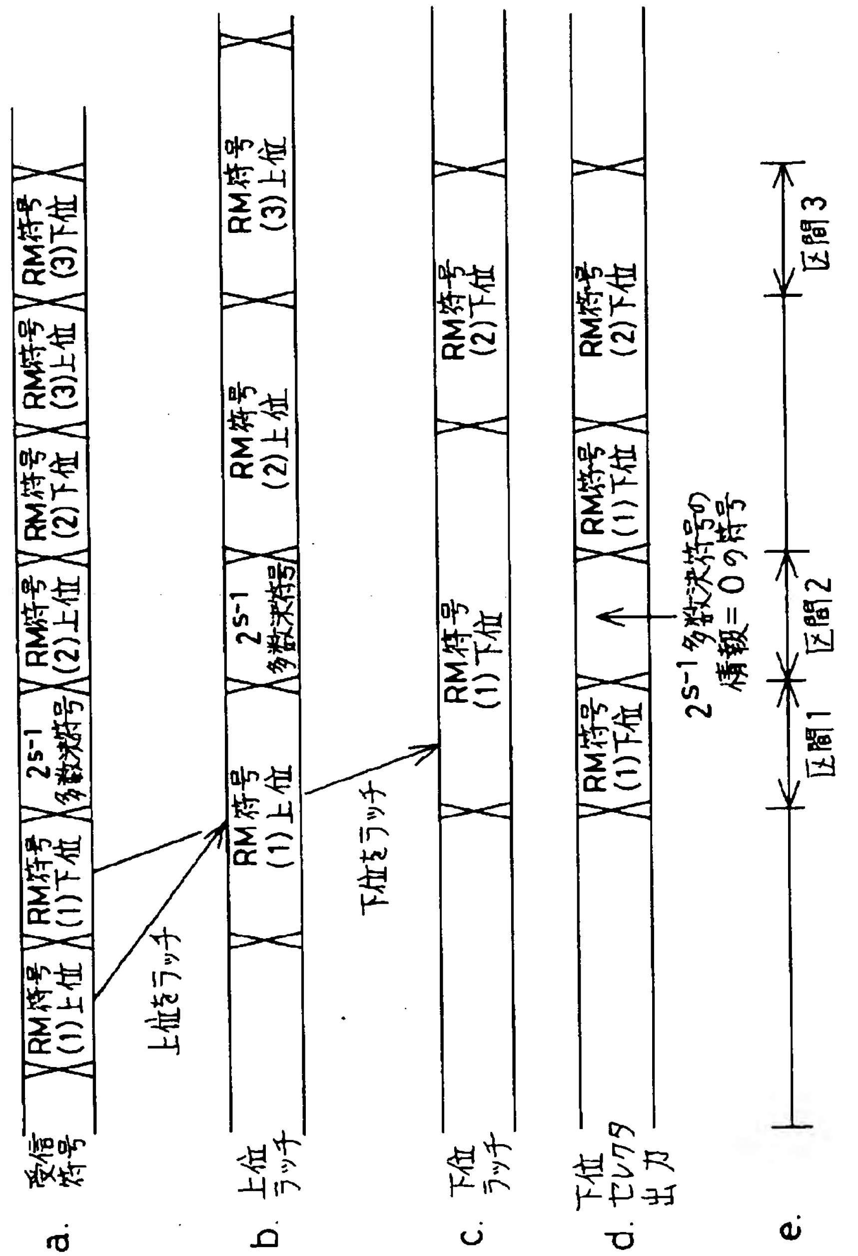
【図9】

2 $^{s-1}$ 多数決符号とリード・マラー符号を復号する基本構成図



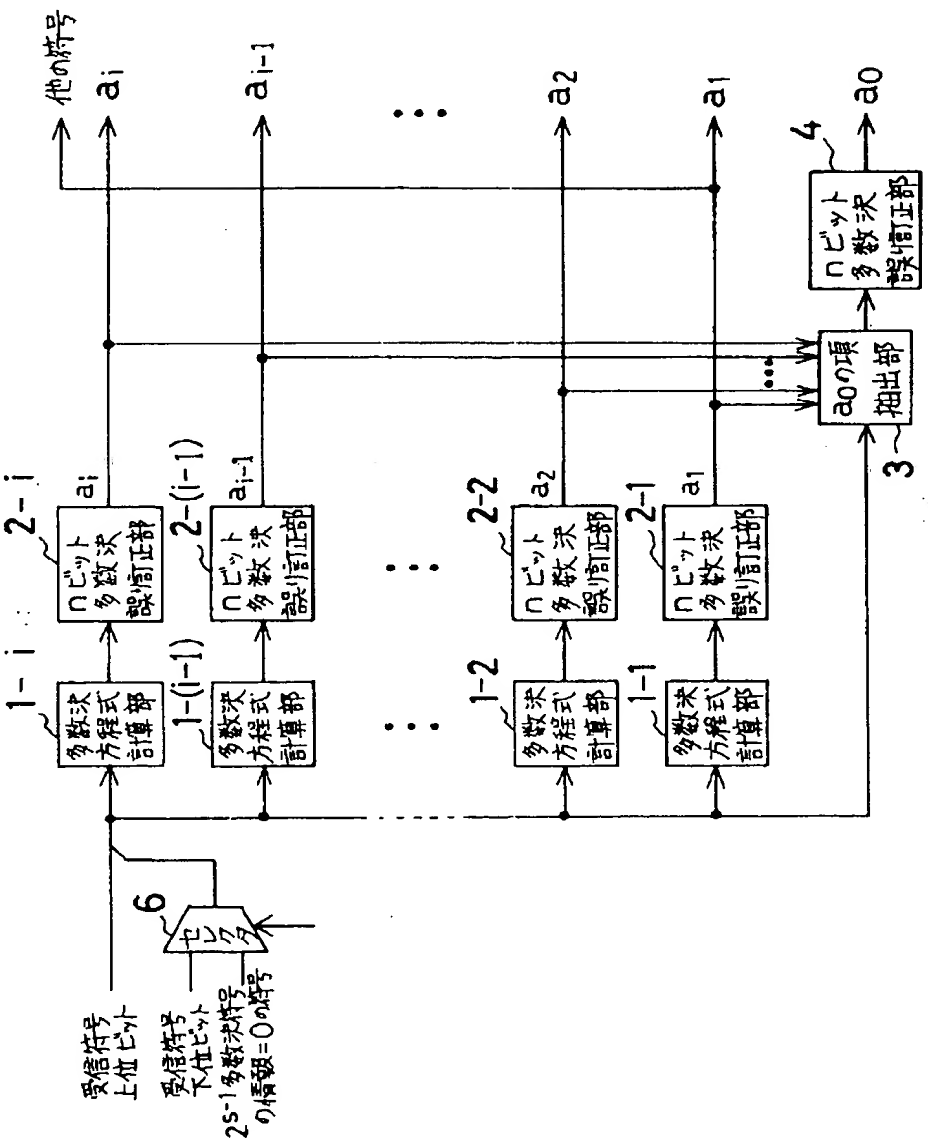
【図10】

図9のタイミングチャート



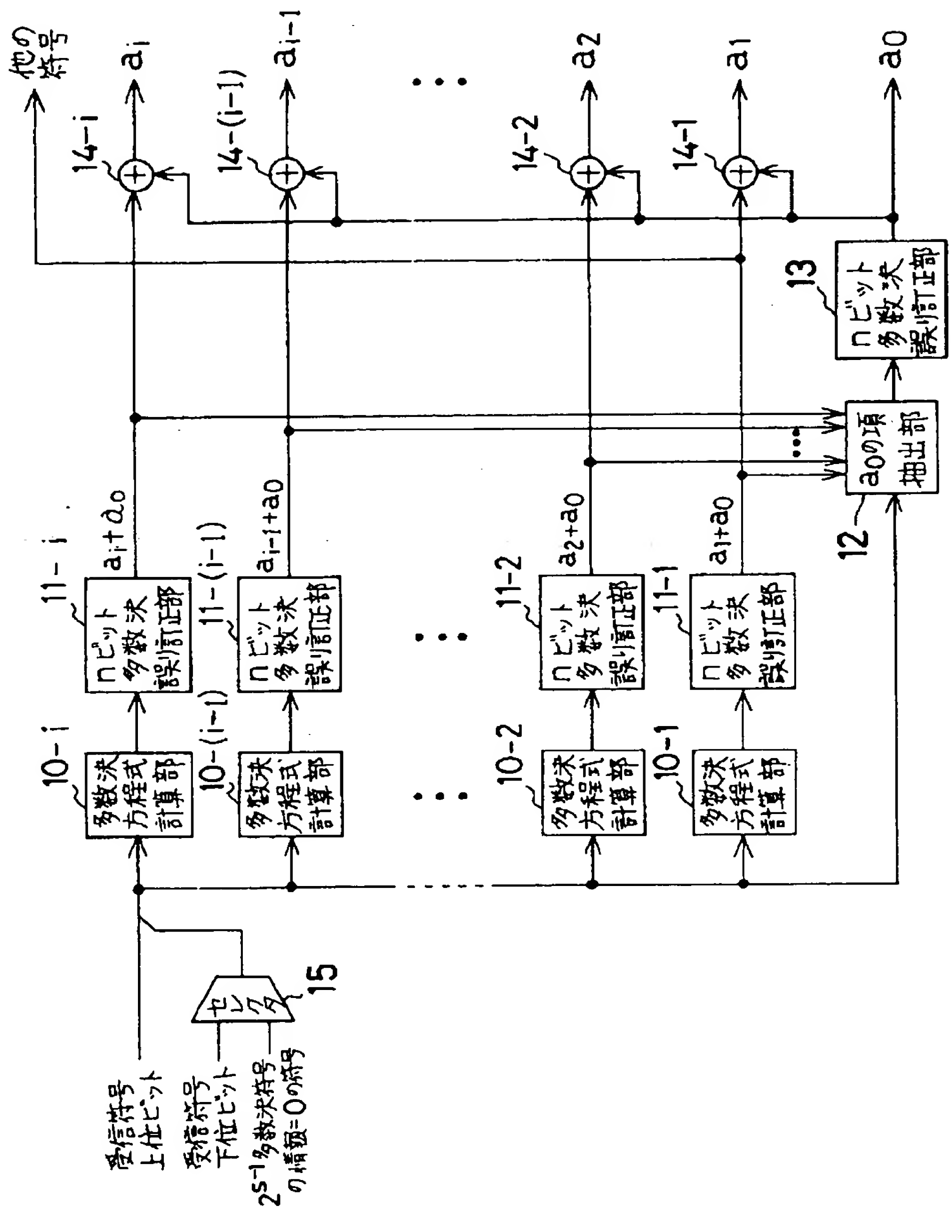
【図11】

2^{s-1}多数決符号の復号を含む復号回路の
実施例1の構成図



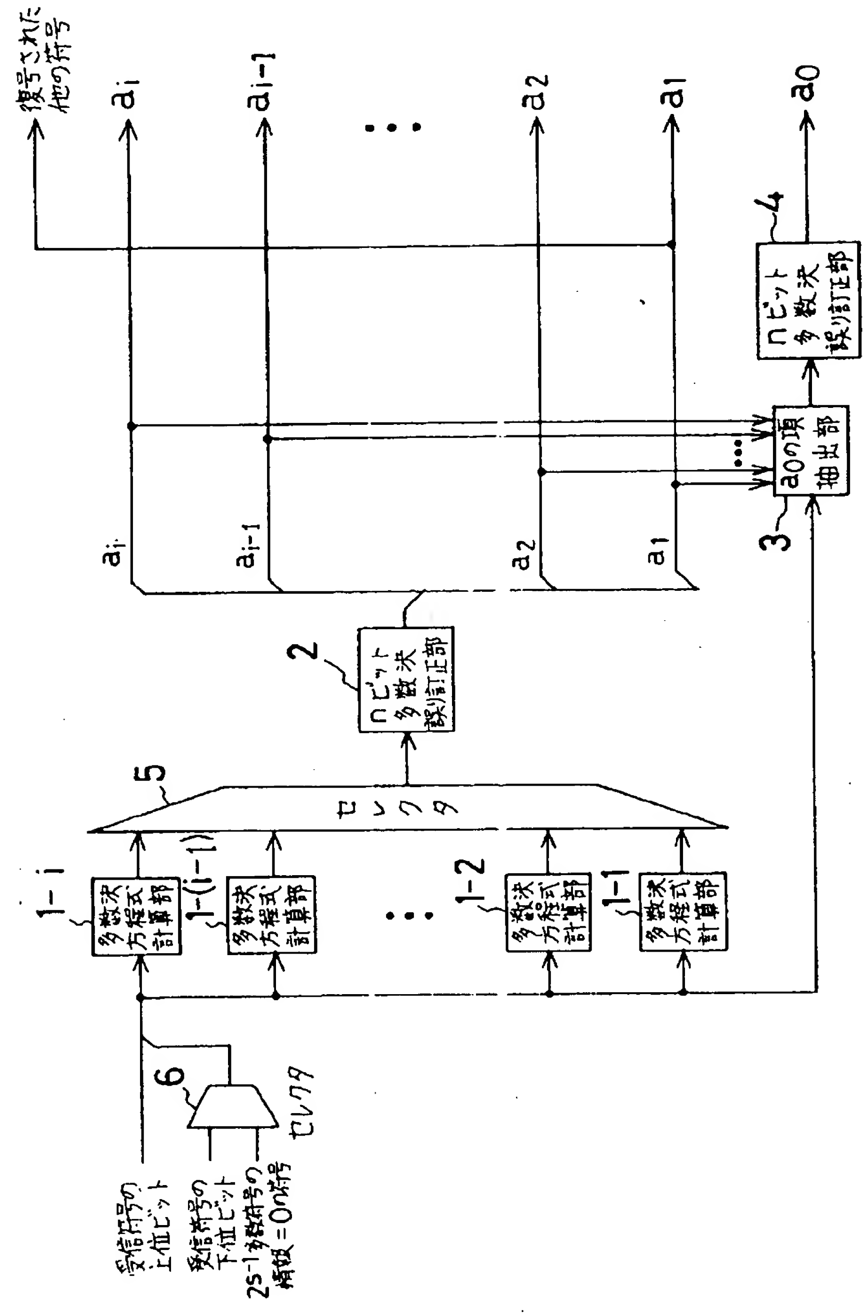
【図１２】

2^s-1 多数決符号の復号を含む復号回路の実施例２の構成図



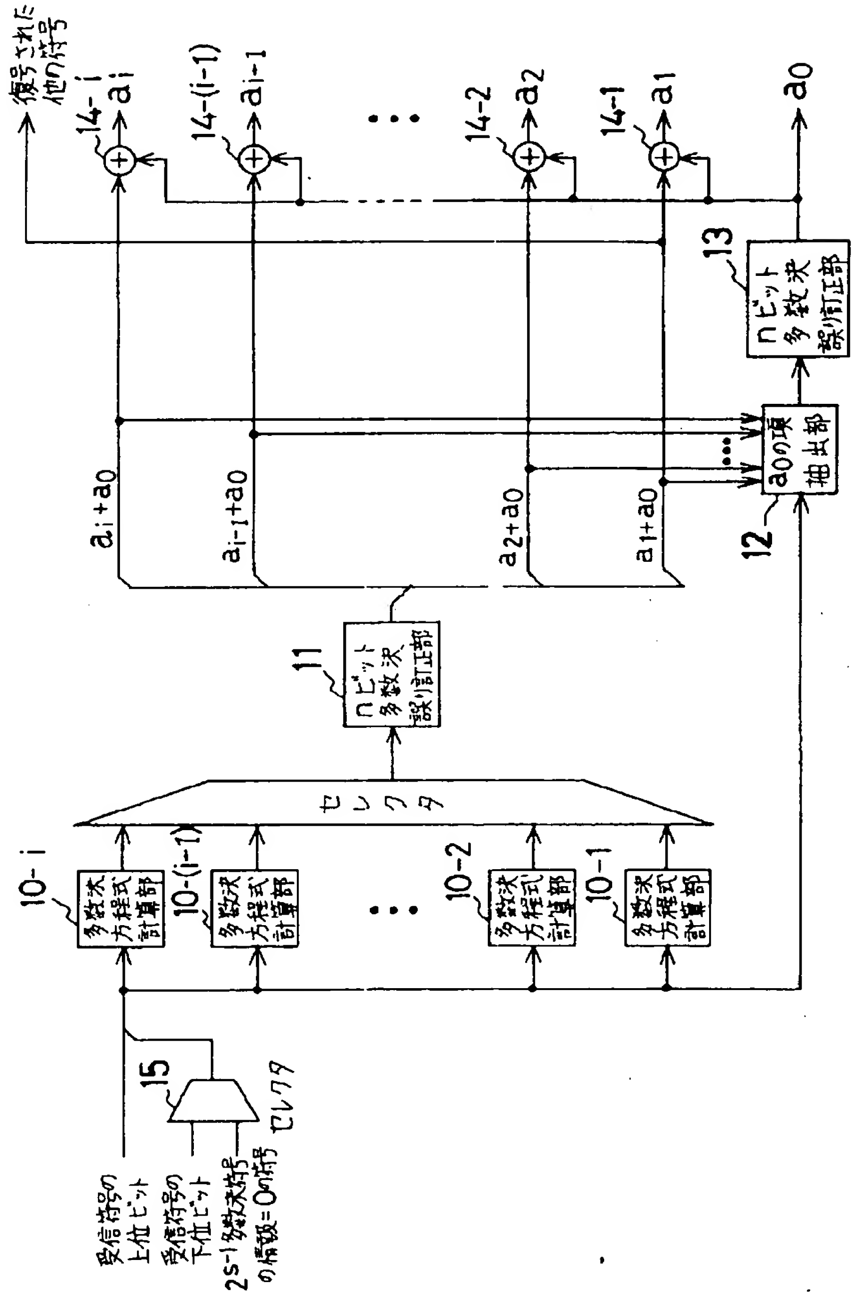
【図１３】

2^{s-1}多数決符号の復号を含む復号回路の実施例３の構成図



【図１４】

2^{s-1} 多数決符号の復号を含む復号回路の実施例４の構成図



フロントページの続き

(72)発明者　篠宮　知宏
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　阿比留　節雄
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　▲廣▼田　正樹
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内

(72)発明者　宮部　正剛
神奈川県川崎市中原区上小田中1015番地
富士通株式会社内